Syst. Biol. 61(4):579–593, 2012

DOI:10.1093/sysbio/syr131
Advance Access publication on January 4, 2012

Phylogenetic Inference via Sequential Monte Carlo

ALEXANDRE BOUCHARD-CÔTÉ[1], SRIRAM SANKARARAMAN[2], AND MICHAEL I. JORDAN[3,*]

[1]*Department of Statistics, University of British Columbia, Vancouver, BC V6T 1Z2, Canada;* [2]*Department of Genetics, Harvard Medical School, Boston, MA 02115, USA; and* [3]*Department of Statistics and Computer Science Division, University of California, Berkeley, CA 94720, USA;*
**Correspondence to be sent to: Department of Statistics and Computer Science Division, University of California, Berkeley, CA 94720, USA;*
E-mail: jordan@cs.berkeley.edu.

Received 18 July 2011; reviews returned 21 August 2011; accepted 18 December 2011
Associate Editor: Mark Holder

Abstract.—Bayesian inference provides an appealing general framework for phylogenetic analysis, able to incorporate a wide variety of modeling assumptions and to provide a coherent treatment of uncertainty. Existing computational approaches to Bayesian inference based on Markov chain Monte Carlo (MCMC) have not, however, kept pace with the scale of the data analysis problems in phylogenetics, and this has hindered the adoption of Bayesian methods. In this paper, we present an alternative to MCMC based on Sequential Monte Carlo (SMC). We develop an extension of classical SMC based on partially ordered sets and show how to apply this framework—which we refer to as *PosetSMC*—to phylogenetic analysis. We provide a theoretical treatment of PosetSMC and also present experimental evaluation of PosetSMC on both synthetic and real data. The empirical results demonstrate that PosetSMC is a very promising alternative to MCMC, providing up to two orders of magnitude faster convergence. We discuss other factors favorable to the adoption of PosetSMC in phylogenetics, including its ability to estimate marginal likelihoods, its ready implementability on parallel and distributed computing platforms, and the possibility of combining with MCMC in hybrid MCMC–SMC schemes. Software for PosetSMC is available at http://www.stat.ubc.ca/ bouchard/PosetSMC. [Bayesian inference; sequential Monte Carlo.]

Although Bayesian approaches to phylogenetic inference have many conceptual advantages, there are serious computational challenges needing to be addressed if Bayesian methods are to be more widely deployed. These challenges are currently being shaped by two trends: (i) Advances in computer systems make it possible to perform iterations of Markov chain Monte Carlo (MCMC) algorithms more quickly than before and (ii) however, thanks to advances in sequencing technology, the amount of data available in typical phylogenetic studies is increasing rapidly. The latter rate currently dominates the former, and, as a consequence, there are increasing numbers of phylogenetic data sets that are out of the reach of Bayesian inference. Moreover, while future improvement in computational performance is expected to come in the form of parallelization, current methods for parallelizing MCMC (Feng et al. 2003; Altekar et al. 2004; Keane et al. 2005; Suchard and Rambaut 2009) have important limitations (see the Discussion section).

Another issue with MCMC algorithms is the difficulty of computing the marginal likelihood, a quantity needed in Bayesian testing (Robert 2001). The naive estimator has unbounded variance (Newton and Raftery 1994), and alternatives, such as thermodynamic integration (Lartillot and Philippe 2006), are nontrivial to implement in the phylogenetic setting (Gelman and Meng 1998; Fan et al. 2011; Xie et al. 2011).

In this paper, we propose an alternative methodology for Bayesian inference in phylogeny that is based on Sequential Monte Carlo (SMC). Although SMC is generally applied to problems in sequential data analysis (Doucet et al. 2001), we develop a generalized form of SMC—which we refer to as *PosetSMC*—that applies directly to phylogenetic data analysis. We present experiments that show that for a large range of phylogenetic models and reconstruction precision levels, PosetSMC is significantly faster than MCMC. Moreover, PosetSMC automatically provides a well-behaved estimate of the marginal likelihood.

There have been earlier attempts to apply classical SMC to inference in tree-structured models (Görür and Teh 2008; Teh et al. 2008), but this work was restricted to models based on the coalescent. There has also been a large body of work applying sequential methods that are closely related to SMC (e.g., importance sampling and approximate Bayesian computation) to problems in population genetics, but this work also focuses on coalescent models (Griffiths and Tavaré 1996; Beaumont et al. 2002; Marjoram et al. 2002; Iorio and Griffiths 2004; Paul et al. 2011). Although the coalescent model is dominant in population genetics, a wider range of prior families is needed in phylogenetic inference. Examples include the Yule process (Rannala and Yang 1996), uniform priors, and a variety of nonclock priors (Huelsenbeck and Ronquist 2001).

The general problem of Bayesian computation in phylogeny involves approximating an integral over a space $\mathcal{T}$ of phylogenetic trees. These integrals are needed for nearly all Bayesian inferences. To motivate the PosetSMC framework for computing these integrals, it is useful to consider the simpler problem of *maximizing* over a space of phylogenetic trees. When maximizing over phylogenies, two broad strategies, or metaalgorithms, are used: *local search* and *sequential search*. The two strategies use radically different methods to compute optimal trees, and they use a different type of state representation.

In local search, the representation maintained at each step of the algorithm is based on a full specification of a phylogenetic tree (which we will call a *full state* or just a *state* for short). Local search starts at an arbitrary state, and at each iteration, states that are nearby the current candidate are evaluated (where "nearby" is defined according to a user-specified metric, e.g., the metric induced by local branch exchanges or by regrafting; Felsenstein 2003). The current candidate is updated to one of these nearby states until a stopping criterion is met. Stochastic annealing is an example of a local search strategy.

In sequential search, the representation maintained at each step of the algorithm is based on a *partial* specification of a phylogenetic tree. A partial specification can take the form of a phylogenetic forest over the observed taxa, for example, where a phylogenetic forest is defined as a set of phylogenetic trees. We call these entities *partially specified states* or *partial states*. Note that there is a dual representation of partial states: A partial state s can be viewed as a *set* of full states, $D(s) \subseteq \mathcal{T}$: the set of full states that satisfy a given constraint. In the previous example where each partial state s is a forest, the dual set $D(s)$ corresponds to the set of all phylogenetic trees t that contain as subtrees all the trees t' of the forest s, with matching branch lengths.

Sequential search starts at the *least partial state* (the partial state whose dual is the set of all full states; i.e., the partial state where each tree in the forest consists of exactly one taxon) and proceeds iteratively as follows: At each step, a set of possible successors of the current partial state is considered. The successor of a state s refers to a new partial state s' with a dual $D(s')$ strictly contained in the dual $D(s)$ of the current partial state s. Second, a promising successor, or set of successors, is taken as the new candidate partial state(s). Neighbor joining is an example of a sequential search strategy, where successors are obtained by merging two trees in a forest, forming a new forest with one less tree in it.

If an infinite computational budget were available, local search strategies would generally be preferred over sequential ones. For example, stochastic annealing is guaranteed under conditions on the annealing rate to approach the optimal solution, whereas neighbor joining will maintain a fixed error. However, since computational time is a critical issue in practice, cheap algorithms such as neighbor joining are often preferred to more expensive alternatives.

We now return to the Bayesian problem of *integration* over the space of trees. Note that MCMC algorithms for integration can be viewed as analogs of the local search strategies used for maximization. What would then be a sequential strategy for integration? This is exactly where SMC algorithms fit. SMC uses partial states that are successively extended until a fully specified state is reached. Many candidates are maintained simultaneously (these candidate partial states are called "particles"), and unpromising candidates are periodically pruned (typically by resampling). Given unbounded computational resources, MCMC eventually outperforms SMC, but for smaller computation times, or for highly parallelized architectures, SMC is an attractive alternative.

SMC algorithms were originally developed in the context of a restrictive class of models known as *state-space models*. While there has been work on extending SMC to more general setups (Moral et al. 2006; Andrieu et al. 2010), this earlier work has been based on the assumption that the target integral is approximated by integrals over product spaces of increasing dimensionality, an assumption incompatible with the combinatorial aspect of phylogenetic trees.

The work of Tom et al. (2010) applies sequential sampling and reweighting methods to phylogenetics, but in the context of pooling results from stratified analyses. In this paper, we construct a single joint tree posterior.

The remainder of the article is organized as follows. In the Background and Notation section, we review some basic mathematical definitions and notation. We then outline the general PosetSMC framework and present theoretical guarantees for PosetSMC in the Theoretical Guarantees section. Results on synthetic and real data are presented in the Experiments section, and we present our conclusions in the Discussion section.

Background and Notation

Before describing PosetSMC, we introduce our notation, which is based on the notation of Semple and Steel (2003). Let X be a set of leaves (observed taxa), T be a random phylogenetic X tree, with values in a measurable space $\mathcal{T}$ (we will consider both ultrametric setups and nonclock setups), and $\mathcal{Y}$, a set of observations at the leaves. For $X' \subset X$, we use the notation $\mathcal{Y}(X')$ for the subset of observations corresponding to a subset X' of the leaves. In rooted trees, we use the terminology *rooted clade* to describe a maximal subset of leaves $X' \subset X$ that are descendant from an internal node. For a rooted tree t, we denote the set of such subsets by $\Sigma_{\mathrm{r}}(t)$. In the unrooted tree case, the set $\Sigma_{\mathrm{u}}(t)$ is the set of blocks in the bipartitions obtained by removing an edge in the tree; that is, $\Sigma_{\mathrm{u}}(t) = \Sigma_{\mathrm{r}}(t') \cup \{X \backslash X' : X' \in \Sigma_{\mathrm{r}}(t')\}$, where t' is an arbitrary rooting of t.

We assume that the trees $t \in \mathcal{T}$ contain positive branch lengths, encoded as maps from subsets of leaves, 2^X, to the nonnegative real numbers. In the rooted case, $b_{\mathrm{r}}(X' ; t)$ is equal to zero if $X' \notin \Sigma_{\mathrm{r}}(t)$ and equal to the length of the edge joining the root of the clade X' to its parent otherwise. In the nonclock case, $b_{\mathrm{u}}(X' ; t)$ is equal to zero if $X' \notin \Sigma_{\mathrm{u}}(t)$ and to the edge corresponding to the bipartition $X'|X \backslash X'$ otherwise.

We denote the unnormalized target posterior measure given $\mathcal{Y}$ by $\pi_{\mathcal{Y}}$ or π for short. (Formally, $\pi_{\mathcal{Y}}{:}\mathcal{F}_{\mathcal{T}} \to [0, \infty)$, where we use $\mathcal{F}_{\mathcal{T}}$ to denote the Borel sigma algebra on $\mathcal{T}$.) The measure π is the product of a prior and likelihood evaluated at the observations, summing over the states at the internal nodes. We assume that it has an unnormalized density $\gamma_{\mathcal{Y}} : \mathcal{T} \to [0, \infty)$. In most models, $\gamma = \gamma_{\mathcal{Y}}$ can be evaluated at any point in polynomial time (Felsenstein 2003), whereas the normalization

$\|\pi\| = \pi(\mathcal{T})$, which is equal to the marginal likelihood $\mathbb{P}(\mathcal{Y})$ by Bayes' theorem, is hard to compute—estimating it is the first of the two goals of the method of the next section.

The second goal was to compute posterior expectations of the form $\mathbb{E}[\phi(T)|\mathcal{Y}]$ for a problem-dependent *test function* $\phi : \mathcal{T} \to \mathbb{R}^c$ for some $c \in \mathbb{N}$. These functions are generally the sufficient statistics needed to compute Bayes estimators. To define Bayes estimators and to evaluate their reconstructions, we will make use of the *partition metric* (which ignores branch lengths), d_{PM}, and the L1 and (squared) L2 metrics, $d_{\mathrm{L1}}, d_{\mathrm{L2}}$, which take branch lengths into account (Bourque 1978; Robinson and Foulds 1981; Kuhner and Felsenstein 1994). We will use the unrooted versions of these metrics (which allows us to measure distance between rooted trees as well by ignoring the rooting information):

$$d_{\mathrm{PM}}(t,t') = |\Sigma_{\mathrm{u}}(t)\Delta\Sigma_{\mathrm{u}}(t')|,$$

$$d_{\mathrm{L1}}(t,t') = \sum_{X' \subset X} |b_{\mathrm{u}}(X';t) - b_{\mathrm{u}}(X';t')|,$$

$$d_{\mathrm{L2}}(t,t') = \sum_{X' \subset X} (b_{\mathrm{u}}(X';t) - b_{\mathrm{u}}(X';t'))^2,$$

where $A\Delta B$ denotes the symmetric difference of sets A and B. Note that a loss function can be derived from each of these metrics by taking the additive inverse. For example, if the objective is to reconstruct a consensus topology with the least partition metric risk, each coordinate of the function, ϕ_i, takes the form of an indicator function on a unrooted clade $X_i \subset X$, $\phi_i(t) = 1[X_i \in \Sigma_{\mathrm{u}}(t)]$.

We will use the notation $\bar{\pi} = \pi/\|\pi\|$ for normalized measures and $\bar{\pi}(\phi)$ as a shorthand for integration of ϕ with respect to $\bar{\pi}$, for example, $\mathbb{E}[\phi(T)|\mathcal{Y}]$ here.

We will also need some concepts from order theory. A *poset* (partially ordered set) $(\mathcal{S}, \prec)$ is a generalization of the familiar order relation $<$. Poset relations $\prec$ satisfy many of the properties found in $<$ (reflexivity, antisymmetry, and transitivity), but they do not require all pairs of elements of $\mathcal{S}$ to be comparable (i.e., there can be $s, s' \in \mathcal{S}$ with neither $s \prec s'$ nor $s' \prec s$). We say that s' *covers* s if $s \prec s'$, and there are no s'' other than s, s' with $s \prec s'' \prec s'$. Finally, we will refer in the following to the (undirected) *Hasse diagram* of a poset, which is an undirected graph where the set of vertices is $\mathcal{S}$, and there is an edge between s and s' whenever s covers s'.

Poset SMC on Phylogenetic Trees

We now turn to the description of the PosetSMC framework. This framework encompasses existing work on tree-based SMC (Teh et al. 2008) as a special case and yields many new algorithms.

Overview

PosetSMC is a flexible algorithmic framework, with the flexibility deriving from two sources. The first is the choice of *proposal*: Just as in MCMC methods, PosetSMC requires the specification of a proposal distribution and there is freedom in this choice. Second, PosetSMC requires an *extension* of the density γ, which is defined on trees, to a density γ_* that is defined on forests.

Figure 1 presents an overview of the overall PosetSMC algorithmic framework. It will be useful to refer to this figure as we proceed through the formal specification of the framework.

We begin by discussing proposal distributions. Note that MCMC algorithms also involve proposal distributions, but, in contrast to MCMC, where the proposals are transitions from $\mathcal{T}$ to $\mathcal{T}$, the proposals in PosetSMC are defined over a larger space, $\mathcal{S} \supset \mathcal{T}$, which we will be endowing with a partial order structure. The elements of this larger space are called partial states, and they have the same dual interpretation as the partial states described in the context of maximization algorithms in the introduction. We denote the dual of s by $D(s)$, where $D : \mathcal{S} \to \mathcal{F}_{\mathcal{T}}$. We write ν_s for the proposal distribution (a regular conditional probability) at s, with density $q(s \to s')$, $s, s' \in \mathcal{S}$, and we let q^n denote the n-step proposal density. Although an MCMC proposal is generally defined using a metric (i.e., sampling is done within a subset of states that are nearby the current state), we need the richer structure of posets to define proposals in our case.

In particular, we assume that the proposal distributions are such that they allow a *poset representation*.

Assumption 1 We assume that there is a poset $(\mathcal{S}, \prec)$ such that $q^n(s \to s') > 0$ for some $n \geqslant 1$ if and only if $s \prec s'$.

The associated poset representation encodes whether states are reachable via applications of the proposal distribution. Note that this puts restrictions on valid proposal distributions: In particular—and in contrast to MCMC proposals—directed cycles should have zero density under q.

Using the proposal distribution, our algorithms iteratively propose successor states s' with $s \prec s'$, until the algorithm arrives at partial states that fully specify a phylogenetic tree, where $|D(s)| = 1$. In order to keep track of the number of proposal applications needed before arriving at a fully specified state, we assume that the posets are equipped with a *rank*, a monotone function $\rho : \mathcal{S} \to \{0, \ldots, R\}$ such that whenever s covers s', $\rho(s) = \rho(s') + 1$. At each proposal step, the rank is increased by one, and the set of states of highest rank R is assumed to coincide with the set of fully specified states: $\rho^{-1}(R) = \mathcal{T}$. At the other extremity of the poset, we let $\perp$ denote the unique minimum element with $D(s) = \mathcal{T}$.

In addition to a proposal distribution, a second object needs to be provided to specify a PosetSMC algorithm: an *extension* $\gamma_* : \mathcal{S} \to [0, \infty)$ of the density γ from $\mathcal{T}$ to $\mathcal{S}$.

In the remainder of the paper, we provide examples of proposals and extensions, and we also provide a precise set of conditions that are sufficient for correctness of a PosetSMC algorithm. Before turning to those results,

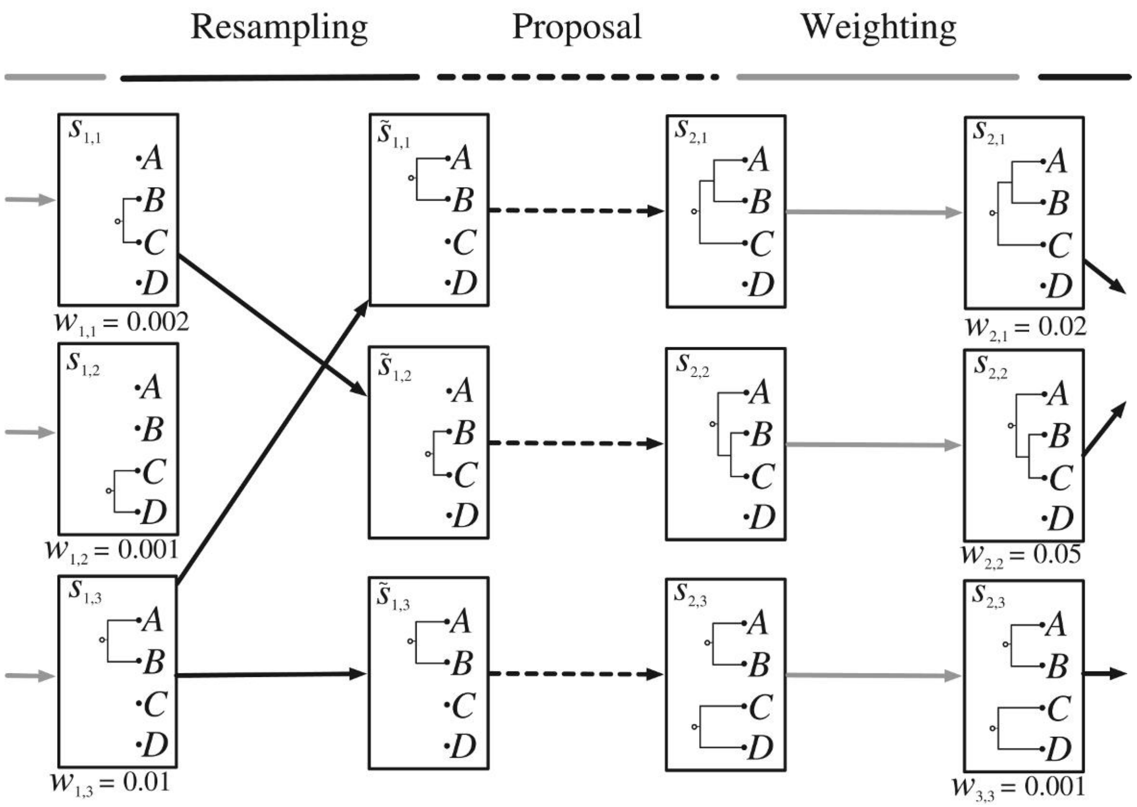

FIGURE 1. An overview of the PosetSMC algorithmic framework. A PosetSMC algorithm maintains a set of partial states (three partial states are shown in the leftmost column in the figure; each partial state is a forest over the leaves A, B, C, and D). Associated with each partial state is a positive-valued weight. The algorithm iterates the following three steps: (i) resample from the weighted partial states to obtain an unweighted set of partial states, (ii) propose an extension of each partial state to a new partial state in which two trees in the forest have been connected, and (iii) calculate the weights associated with the new partial states.

we give a simple concrete example of a proposal distribution in the case of ultrametric trees.

In an ultrametric setup, the examples we consider are based on $\mathcal{S}$ defined as the set of *ultrametric forests* over X. An ultrametric forest $s = \{(t_i, X_i)\}$ is a set of ultrametric X_i-trees t_i such that the disjoint union of the leaves yields the set of observed taxa: $\sqcup X_i = X$. The rank of such a forest is defined as $|X| - |s|$.

Defining the *height* of an ultrametric forest as the height of the tallest tree in the forest, we can now introduce the partial order relationship we use for ultrametric setups. Let s and s' be ultrametric forests. We deem that $s \prec s'$ if all the trees in s appear as subtrees in s' with matching branch lengths and if the height of s' is strictly greater than the height of s. As we will see shortly, any proposal that simply merges a pair of trees and strictly increases the forest height is a valid proposal.

Algorithm Description

Once these two ingredients are specified—a proposal and an extension—the algorithm proceeds as follows. At each iteration r, we assume that a list of K partial states is maintained (each element of this list is called a *particle*). These particles are denoted by $s_{r,1}, \ldots, s_{r,K} \in \mathcal{S}$. We also assume that there is a positive *weight* $w_{r,k}$ associated with each particle $s_{r,k}$. Combined together, these form an *empirical measure*:

$$\pi_{r,K}(\cdot) = \sum_{k=1}^{K} w_{r,k} \delta_{s_{r,k}}(\cdot), \tag{1}$$

where $\delta_x(A) = 1$ if $x \in A$ and 0 otherwise.

Initially, $s_{0,k} = \bot$ and $w_{0,k} = 1/K$ for all k. Given the list of particles and weights from the previous iteration $r-1$, a new list is created in three steps. The first step can be understood as a method for pruning unpromising particles. This is done by sampling independently K times from the normalized empirical distribution $\bar{\pi}_{r-1,K}$. The result of this step is that some of the particles (mostly those of low weight) will be pruned. (Other sampling schemes, such as stratified sampling and dynamic on-demand resampling, can be used to further improve performance; see Doucet et al. 2001.) We denote the sampled particles by $\tilde{s}_{r-1,1}, \ldots, \tilde{s}_{r-1,K}$. The second step is to create new particles, $s_{r,1}, \ldots, s_{r,K}$, by extending the partial states of each of the sampled particles from the previous iteration. This is done by sampling K times from the proposal distribution, $s_{r,k} \sim \nu_{\tilde{s}_{r-1,k}}$. The third step is to compute weights for the new particles:

$$w_{r,k} = \frac{\gamma_*(s_{r,k})}{\gamma_*(\tilde{s}_{r-1,k})q(\tilde{s}_{r-1,k} \to s_{r,k})}. \quad (2)$$

Finally, the target distribution is approximated by $\bar{\pi}_{R,K}$, so that the target conditional expectation $\bar{\pi}(\phi)$ is approximated by $\bar{\pi}_{R,K}(\phi)$. As for the marginal likelihood, the estimate is given by the product over ranks of the weight normalizations:

$$\|\pi\| \approx \prod_{r=1}^{R} \frac{1}{K}\|\pi_{r,K}\|. \quad (3)$$

It is worth highlighting some of the similarities and differences between PosetSMC and other sampling-based algorithms in phylogenetics, in particular the nonparametric bootstrap and MCMC algorithms. First, as in the case of the bootstrap, the K particles in PosetSMC are sampled with replacement, and the number of particles (which remains constant throughout the run of the algorithm) is a parameter of the algorithm (see the Discussion section for suggestions for choosing the value of K). Second, new partial states obtained from the proposal distribution are always "accepted" (as opposed to Metropolis–Hastings, where some proposals are rejected). Third, the weights of newly proposed states influence the chance each particle survives into the next iteration. Fourth, once full states have been created by PosetSMC, the algorithm terminates. Finally, PosetSMC is readily parallelized, simply by distributing particles across multiple processors. MCMC phylogenetic samplers can also be parallelized, but the parallelization is less direct (see the Discussion section for further discussion of this issue).

Theoretical Guarantees

In this section, we give theoretical conditions for statistical correctness of PosetSMC algorithms. More precisely, we provide sufficient conditions for consistency of the marginal likelihood estimate and of the target expectation as the number of particles K goes to infinity.

The sufficient conditions are as follows. (Note that all of them have an intuitive interpretation and are easy to check.) The first group of conditions concern the proposal:

Assumption 2 Let $q : \mathcal{S} \to \mathcal{S}$ be a proposal density, with associated poset $(\mathcal{S}, \prec)$ as defined in the previous section. We assume that the (undirected) Hasse diagram corresponding to $(\mathcal{S}, \prec)$ is (a) connected and (b) acyclic.

Assumption 2(a) can be compared with an irreducibility condition in MCMC theory: There must be a path of positive proposal density reaching each state. Assumption 2(b) is more subtle but is very important in our framework. It can be restated as saying that for each partial state s, there should be at most one sequence of partial states connecting it to the initial state $\perp = s_0, \ldots, s_n = s$, with $q(s_i \to s_{i+1}) > 0$ for all i. This insures that trees are not overcounted.

Next, we introduce the conditions on the extension γ_*:

Assumption 3 The density γ_* is (a) positive, $\gamma_*(s) > 0$ for all $s \in \mathcal{S}$, and (b) it extends γ, in the sense that there is a $C > 0$ with $\gamma = 1_{\mathcal{T}} C \gamma_*$.

Note that we do not require that it be feasible to compute C—its value is not needed in our algorithms. Assumption 3(a) insures that the ratio in Equation (2) is well defined, and Assumption 3(b) is the step where the form of the target density γ is taken into account.

Under these assumptions, and assuming regularity conditions on ϕ and γ_*, we have that PosetSMC is consistent (see Appendix 1 for the proof):

Proposition 4 If Assumptions 1, 2, and 3 are met, and γ_* is bounded, then as $K \to \infty$,

$$\frac{1}{K^R}\prod_{r=1}^{R}\|\pi_{r,K}\| \overset{K\to\infty}{\longrightarrow} \|\pi\|, \quad (4)$$

and moreover, when $\bar{\pi}(|\phi|) < \infty$,

$$\bar{\pi}_{R,K}(\phi) \overset{K\to\infty}{\longrightarrow} \bar{\pi}(\phi). \quad (5)$$

Examples

In this section, we provide several examples of proposal distributions and extensions that meet the conditions described in the previous section.

Proposals

In the ultrametric case, we have given in the Overview section a recipe for creating valid proposal distributions. In particular, we can use proposals that merge pairs while strictly increasing the height of the forest. We begin this section by giving a more detailed explanation of why the strict increase in height is important and how it solves an overcounting issue.

To understand this issue, let us consider a counterexample with a naive proposal that does not satisfy Assumption 2(b) and show that it leads to a biased estimate of the posterior distribution. For simplicity, let us consider ultrametric phylogenetic trees with a uniform prior with unit support on the time between speciation events.

There are $(2n-3)!! = 1 \times 3 \times 5 = 15$ rooted topologies on four taxa, and in Figure 2, we show schematically a subset of the Hasse diagram of the particle paths that lead to these trees under the naive proposal described above. It is clear from the figure that a balanced binary tree on four taxa, with rooted clades $\{A, B\}, \{C, D\}$, can be proposed in two different ways: either by first merging A and B, then C and D, or by merging C and D, then A and B. On the other hand, an unbalanced tree on the same set of taxa can be proposed in a single way, for example, $\{A, B\} \to \{A, B, C\}$. As a consequence, if we

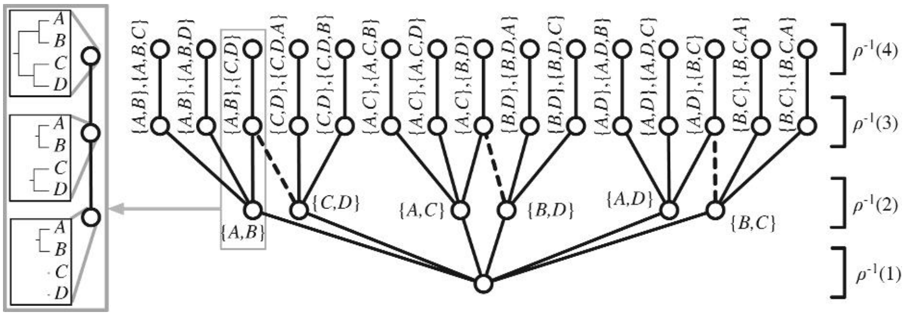

FIGURE 2. To illustrate how PosetSMC sequentially samples from the space of trees, we present a subset of the Hasse diagram induced by the naive proposal described in the Examples section. Note that this diagram is not a phylogenetic tree: The circles correspond to partial states (phylogenetic forests), organized in rows ordered by their rank ρ, and edges denote a positive transition density between pairs of partial states. The forests are labeled by the union of the sets of nontrivial rooted clades over the trees in the forest. The dashed lines correspond to the proposal moves forbidden by the strict height increase condition (Assumption 2(b) in the text). Note that we show only a subset of the Hasse graph since the branch lengths make the graph infinite. The subset shown here is based on an intersection of *height function fibers*: Given a subset of the leaves $X' \subset X$, we define the height function $h_{X'}(s)$ as the height of the most recent common ancestor of X' in s, if X' is a clade in one of the trees in s, and ω otherwise, where $\omega \notin \mathbb{R}$. Given a map $f : 2^X \to [0,\infty)$, the subset of the vertices of the Hasse diagram shown is given by $\cap_{X' \subset X} h_{X'}^{-1}(\{f(X'), \omega\})$. The graph shown here corresponds to any f such that $f(\{A,B\}) < f(\{C,D\})$, $f(\{A,C\}) < f(\{B,D\})$, and $f(\{A,D\}) < f(\{B,C\})$.

assume there are no observations at the leaves, the expected fraction of particles with a caterpillar topology of each type is 1/18 (because there are 18 distinct paths in Fig. 2), whereas the expected fraction of particles with a balanced topology of each type is $2/18 = 1/9$. However, since we have assumed there are no observations, the posterior should be equal to the prior, a uniform distribution. Therefore, the naive proposal leads to a biased approximate posterior.

The strict height increase condition incorporated in our proposal addresses this issue. The dashed lines in Figure 2 show which naive proposals are forbidden by the height increase condition. After this modification, the bias disappears:

PROPOSITION 5 Proposals over ultrametric forests that merge one pair of trees while strictly increasing the height of the forest satisfy Assumption 2.

Proof. It is enough to show that each $s \in \mathcal{S}$ covers at most one $s' \in \mathcal{S}$. If $s = \perp$, this holds trivially. If $s \neq \perp$, there is a unique s' covered by s, given by splitting the unique tallest tree in the forest (removing the edges connected to the root). □

The proposals used in Teh et al. (2008) all fall in this category. For example, for the "PriorPrior" proposal ν_s, a pair of trees in the forest s is sampled uniformly at random, and the height increment of the forest is sampled from an exponential with rate $\binom{|s|}{2}$, the prior waiting time between two coalescent events.

Again, many other options are available. For example, even when the prior is the coalescent model, one may want use a proposal with fatter tails to take into account deviation from the prior brought by the likelihood model. One way to achieve this is the "PostPost" proposal discussed by Teh et al. (2008), where local posteriors are used for both the height increment and the choice of trees to coalesce (see Appendix 2 for further discussion of this proposal). That approach has some drawbacks, however; it is complex to implement and is only applicable to likelihoods obtained from Brownian motion. Simpler heavy-tailed proposal distributions may be useful.

Proposals can also be informed by heuristics H as discussed in the next section. This can be done, for example, by giving higher proposal density to pairs of trees that form a subtree in $H(s)$.

In the nonclock case, we let $\mathcal{S}$ be the set of rooted nonclock forests over X. A nonclock forest, $s = \{(t_i, X_i)\}$, is a set of nonclock X_i-trees t_i such that the disjoint union of the leaves consists in the set of observed taxa, $\sqcup X_i = X$. Defining the *diameter* of a rooted forest as twice the maximum distance between a leaf and a root over all trees in the forest, we get that any proposal that merges a pair of trees and strictly increases the forest diameter is a valid proposal. The unique state covered by $s \neq \perp$ is the one obtained by splitting the tree with the largest diameter.

Extensions

We turn to the specification of extensions, γ_*, of the density γ from $\mathcal{T}$ to $\mathcal{S}$. There is a simple recipe for extensions that works for both nonclock and ultrametric trees: Given a posterior distribution model $\pi_{\mathcal{Y}}$, set the extension over a forest $s = \{(t_i, X_i)\}$ to be equal to

$$\gamma_*(s) = \prod_{(t_i,X_i)\in s} \gamma_{\mathcal{Y}(X_i)}(t_i). \qquad (6)$$

We call this extension the *natural forest extension*. This definition satisfies Assumption 3 by construction.

More sophisticated possibilities exist, with different computational trade-offs. For example, it is possible to connect the trees in the forest on the fly, by using a fast heuristic such as neighbor joining (Saitou and Nei 1987). If we let $H : \mathcal{S} \to \mathcal{T}$ denote this heuristic, we then get this alternate extension:

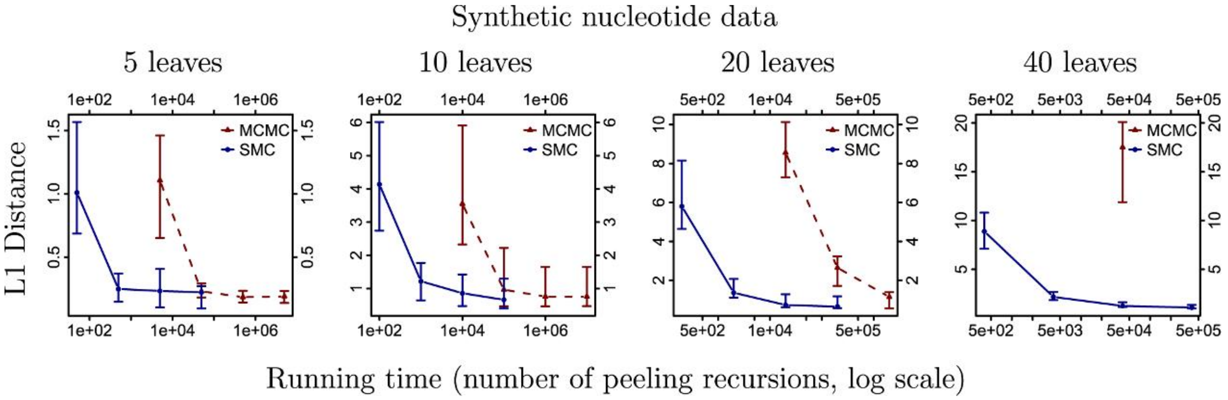

FIGURE 3. Comparison of the convergence time of PosetSMC and MCMC. We generated coalescent trees of different sizes and data sets of 1000 nucleotides. We computed the L1 distance of the minimum Bayes risk reconstruction to the true generating tree as a function of the running time (in units of the number of peeling recursions, on a log scale). The missing MCMC data points are due to MrBayes stalling on these executions.

$$\gamma_*(s) = \gamma(H(s)).$$

As long as all the trees in s appear as subtrees of $H(s)$, this definition also satisfies Assumption 3. This approach can also be less greedy, by taking into account the effect of the future merging operations. We present other examples of extensions in Appendix 2.

EXPERIMENTS

In this section, we present experiments on real and synthetic data. We first show that for a given error level, SMC takes significantly less time to converge to this error level than MCMC. For the range of tree sizes considered (5–40), the speed gap of SMC over MCMC was around two orders of magnitude. Moreover, this gap widens as the size of the trees increases. We then explore the impact of different likelihoods, tree priors, and proposal distributions. Finally, we consider experiments with real data, where we observe similar gains in efficiency as with the simulated data.

Computational Efficiency

We compared our method with MCMC, the standard approach to approximating posterior distributions in Bayesian phylogenetic inference (see Huelsenbeck et al. 2001 for a review). We implemented the PosetSMC algorithm in Java and used MrBayes (Huelsenbeck and Ronquist 2001) as the baseline MCMC implementation. A caveat in these comparisons is that our results depend on the specific choice of proposals that we made.

In the experiments described in this section, we generated 40 trees from the coalescent distribution of sizes $\{5, 10, 20, 40\}$ (10 trees of each size). For each tree, we then generated a data set of 1000 nucleotides per leaf from the Kimura two-parameter model (K2P) (Kimura 1980) using the Doob–Gillespie algorithm (Doob 1945). In this section, both the PosetSMC and MCMC algorithms were run with the generating K2P model and coalescent prior, fixing the parameters. We use the PriorPrior proposal as described in Teh et al. (2008). PriorPrior chooses the trees to merge and the diameter of the new state from the prior; that is, the trees are chosen uniformly over all pairs of trees, whereas the new diameter is obtain by adding an appropriate exponentially distributed increment to the old diameter. We consider bigger trees as well as other proposals and models in the next sections.

For each data set, we ran MCMC chains with increasing numbers of iterations from the set $\{10^3, 10^4, 10^5, 10^6\}$. We also ran PosetSMC algorithms with increasing numbers of particles from the set $\{10^1, 10^2, 10^3, 10^4\}$. Each experiment was repeated 10 times, for a total of 3200 executions.

We computed consensus trees from the samples and measured the distance of this reconstruction to the generating tree (using the metrics defined in the Background and Notation section). The results are shown in Figure 3 for the L1 metric. For each algorithm setting and tree size, we show the median distance across 100 executions, as well as the first and third quartiles. A speedup of over two orders of magnitudes can be seen consistently across these experiments.

In both the PosetSMC and MCMC algorithms, the computational bottleneck is the peeling recurrence (Felsenstein 1981), which needs to be computed at each speciation event in order to evaluate $\gamma(t)$. Each call requires time proportional to the number of sites times the square of the number of characters (this can be accelerated by parallelization, but parallelization can be implemented in both MCMC and PosetSMC samplers and thus does not impact our comparison). We therefore report running times as the number of times the peeling recurrence is calculated. As a sanity check, we also did a controlled experiment on real data in a single user pure Java setting, showing similar gains in wall clock time. (These results are presented in the Experiments on Real Data section).

In Figures 4 and 5, we show results derived from the series of experiments described above for other metrics. Note, in particular, the result in Figure 4; we see that for a fixed computational budget, the gap between PosetSMC and MCMC increases dramatically as the size of the tree increases.

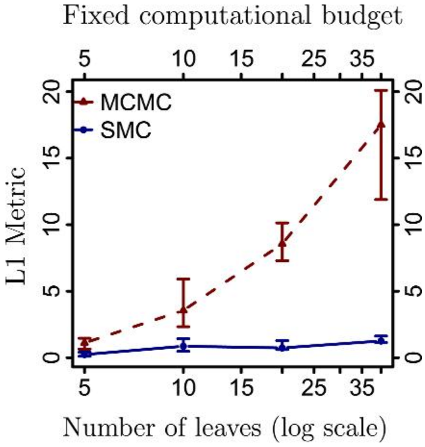

FIGURE 4. L1 distances of the minimum Bayes risk reconstruction to the true generating tree (averaged over trees and executions) as a function of the tree size (number of leaves on a log scale), measured for SMC algorithms run with 1000 particles and MCMC algorithms run for 1000 iterations.

Varying Proposals, Priors, and Likelihoods

In this section, we explore the effect of changing proposals, priors, and likelihood models. We also present results measured by wall clock times.

We first consider data generated from coalescent trees. The proposal distribution is used to choose the trees in a partial state that are merged to create a new tree (in the successor partial state) as well as the diameter of the new state. In Figure 6, we compare two types of proposal distributions, PriorPrior, described in the previous section, and PriorPost (Teh et al. 2008). PriorPost chooses the diameter of the state from the prior and then chooses the pair of trees to merge according to a multinomial distribution with parameters given by the likelihoods of the corresponding new states. (We provide further discussion of this proposal and PriorPrior in Appendix 2.) Although these proposals were investigated experimentally in Teh et al. (2008), their running times were not compared in a satisfactory manner in that work. In particular, the running time was estimated by the number of particles. This is problematic since for a given binary X-tree, PriorPost uses $O(|X|^3)$ peeling recurrences per particle, whereas PriorPrior uses only $O(|X|)$ recurrences per particle. Since we measure running time by the number of peeling recurrences, our methodology does not have this problem. Surprisingly, as shown in Figure 6, PriorPrior outperforms the more complicated PriorPost by one order of magnitude. We believe that this is because PriorPost uses a larger fraction of its computational budget for the recent speciation events compared with PriorPrior, whereas the more ancient speciation events may require more particles to better approximate the uncertainty at that level. (e.g., suppose, we have a budget of $O(|X|^3)$ peeling recurrences. A PriorPrior sampler can use $O(|X|^2)$ particles and leverages $O(|X|^2)$ peeling recurrence for proposing the top branch lengths. A PriorPost sampler can use only $O(1)$ particles and therefore uses only $O(1)$ peeling recurrences for proposing the top branch lengths.)

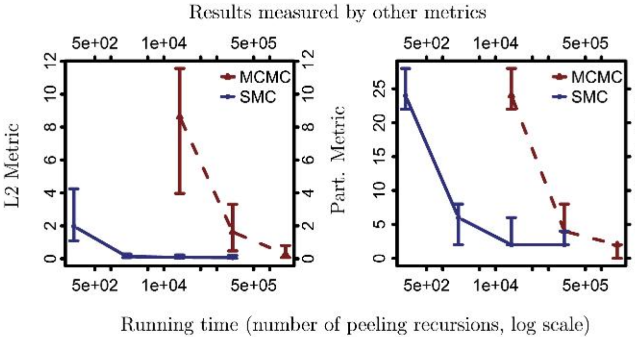

FIGURE 5. Analysis of the same data as in Figure 3, for 20 leaves, but with different metrics: L2 and Partition metrics, respectively.

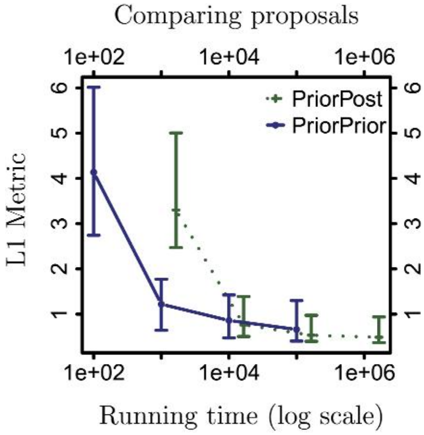

FIGURE 6. Comparison of two types of SMC proposal distributions.

Figure 7 shows the results for data generated by Yule processes (Rannala and Yang 1996) and uniform-branch-length trees. We see that PosetSMC is superior to MCMC in these settings as well.

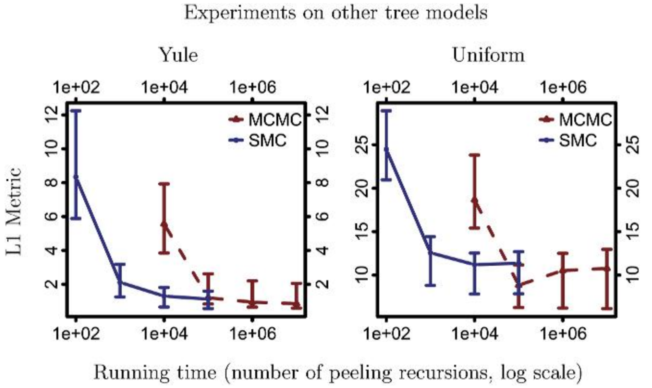

FIGURE 7. Experiments with trees generated from different models. We consider data generated by Yule processes and uniform-branch-length trees. We compare the L1 distance of the minimum Bayes reconstruction with the true generating tree for PosetSMC and MCMC.

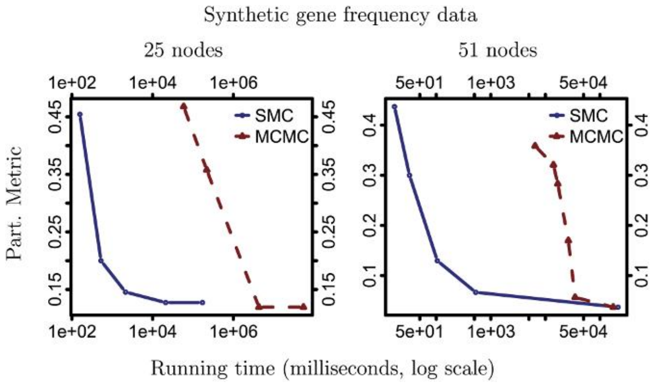

FIGURE 8. Experiments on synthetic gene frequencies using a Brownian motion likelihood model. We show results for two tree sizes. In each case, we plot the partition metric as a function of the wall time in milliseconds, shown on a log scale

Next, we did experiments using a different type of data: synthetic gene frequency. We used Brownian motion to generate frequencies, based the likelihood function on the same model as before and a coalescent prior. As a baseline, we used an MCMC sampler based on standard proposals (Lakner et al. 2008) (stochastic nearest neighbor interchange, global and local multiplicative branch rescaling), with four tempering chains (Neal 1996). Our implementations of continuous-character MCMC and SMC use the same code for computing the likelihood. We verified this code by comparing likelihood values against CONTML (Felsenstein 1973). Since computing the peeling dynamic program is the computational bottleneck for both SMC and MCMC, it is meaningful to compare wall clock times.

The results are shown in Figure 8. In both experiments, there is again a wide computational regime where SMC outperforms MCMC.

Experiments on Real Data

We also tested our algorithm on a comparative RNA data set (Cannone et al. 2002) containing hand-aligned ribosomal RNA. We sampled 16S components from the three domains of life at random to create multiple data sets.

We use the log likelihood of the consensus tree to evaluate the reconstructions. Since finding states of high density is necessary but not sufficient for good posterior approximation, this provides only a partial picture of how well the samplers performed. Since the true tree is not known, however, this gives a sensible surrogate.

We show the results in Figure 9. As in the synthetic data experiments, we found that the PosetSMC sampler required around two orders of magnitude less time to converge to a good approximation of the posterior. Moreover, the advantage of PosetSMC over MCMC becomes more pronounced as the number of taxa increases. For large numbers of iterations and particles, the MCMC sampler slightly outperformed the PosetSMC sampler on the real data we used.

We also performed experiments on frequency data from the Human Genome Diversity Panel. In these experiments, we subsampled 11,511 Single Nucleotide Polymorphisms to reduce site correlations, and we used the likelihood model based on Brownian motion described in the previous section. We show the results in Figure 10, using the log likelihood of the consensus tree as an evaluation surrogate. This shows once again the advantages of SMC methods. To give a qualitative idea of what the likelihood gains mean, we show in Figure 11 the consensus tree from 10,000 MCMC iterations versus the consensus tree from 10,000 PosetSMC particles (the circled data points). Since both runs are under sampled, the higher-order groupings are incorrect in both trees, but we can see that more mid- and low-order ethnic/geographic groupings are already captured by SMC. Incidentally, the position of the circled data points show that in practice, K SMC particles are cheaper to compute than K MCMC iterations. This is because fewer memory writes are necessary in the former case.

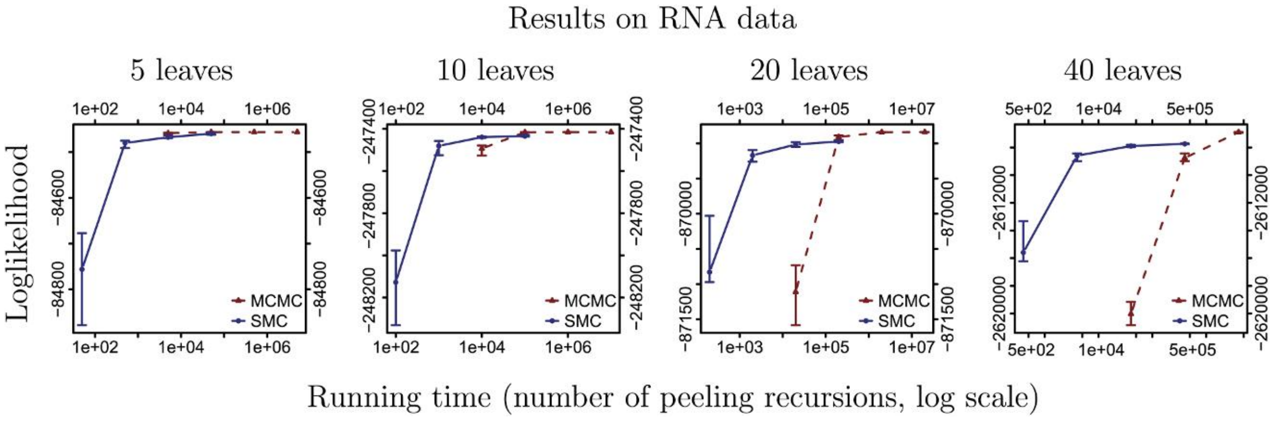

FIGURE 9. Results on ribosomal RNA data (Cannone et al. 2002) on different tree sizes, comparing the log likelihood of the minimum Bayes risk reconstruction from SMC and MCMC approximations, as a function of the running time (in units of the number of peeling recursions on a log scale).

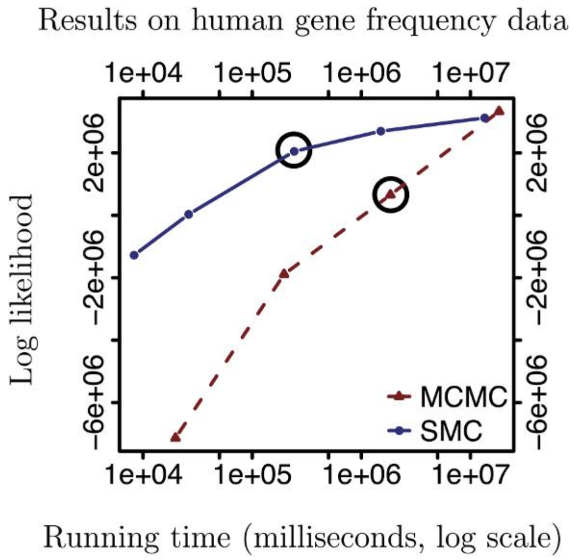

FIGURE 10. Results on human gene frequency data (Li et al. 2008), comparing the log likelihood of the minimum Bayes risk reconstruction from SMC and MCMC approximations, as a function of the running time (in milliseconds, on a log scale).

DISCUSSION

We have presented a general class of SMC methods that provide an alternative to MCMC for Bayesian inference in phylogeny. By making use of the order-theoretic notion of a poset, our PosetSMC methods are able to exploit the recursive structure of phylogenetic trees in computing Bayesian posteriors and marginal likelihoods. Experimental results are quite promising, showing that PosetSMC can yield significant speedups over MCMC; moreover, the relative improvement over MCMC appears to increases as the number of taxa increases, even for simple proposal distributions.

Although we have used simple likelihoods in our experiments, it is also possible to incorporate state-of-the-art models in our framework. Whenever the likelihood for a given tree can be computed exactly and efficiently in a bottom-up manner, our framework applies directly. This includes, for example, models based on more sophisticated rate matrices such as the generalized time-reversible model (Tavaré 1986), covarion models (Penny et al. 2001), approximate context-dependent models (Siepel and Haussler 2004), and fixed-alignment insertion-deletion-aware models (Rivas 2005).

It is also possible to accommodate models that require alternating sampling of parameters and topology, such as random rate matrix and relaxed-clock models (Thorne et al. 1998; Huelsenbeck et al. 2000; Drummond et al. 2006) and insertion–deletion models with alignment resampling (Redelings and Suchard 2005). This can be done by making use of the Particle Markov chain Monte Carlo (PMCMC) method introduced in Andrieu et al. (2010). PMCMC is a hybrid of MCMC and SMC in which an SMC algorithm is used as a proposal in an MCMC chain. Remarkably, it is possible to compute the acceptance ratio for this complex proposal exactly, directly from the output of SMC algorithm. Each factor in the acceptance ratio is computed from the data likelihood estimator of Equation (3): If we denote the estimate of Equation (3) for the current and previous MCMC iteration by L' and L, respectively, then the acceptance ratio is simply $\min\{1, L'/L\}$.

As alluded to earlier, an important advantage of SMC over MCMC is the ease in which it can adapt to parallel architectures. Indeed, the peeling recurrence when computing the proposals is the bottleneck in PosetSMC, and this computation can be easily parallelized by distributing particles across cores. Distribution across clusters is also possible. Note that by exchanging particles across machines after computing the resampling step, it is possible to avoid moving around many particles. In contrast, parallelization of MCMC phylogenetic samplers is possible (Feng et al. 2003; Altekar et al. 2004; Keane et al. 2005; Suchard and Rambaut 2009) but is not as direct. Moreover, all three main approaches to MCMC parallelization have limitations: The first method is to parallelize the likelihood computation, including Graphics Processing Unit parallelization; when scale comes from a large number of taxa, however, this approach reaches its limits. The second method involves running several independent chains; however, this approach suffers from a lack of communication across processors: If one processor finds a way of getting out of a local optimum, it has no way of sharing this information with the other nodes. The third method is to augment the MCMC chain using auxiliary variables, for example, by using parallel tempering (Swendsen and Wang 1986), or multiple-try Metropolis–Hastings (Liu et al. 2000). However, these state augmentations induce tradeoffs between computational efficiency and statistical efficiency.

A practical question in the application of PosetSMC is the choice of K, the number of particles. Although this choice depends on the specific test functions that are being estimated, one generic measure of the accuracy of the PosetSMC estimator can be based on the effective sample size (ESS) (Kong et al. 1994)—the number of independent samples from the true posterior required to obtain an estimator with the same variance as the PosetSMC estimator. A small ESS (relative to K) is indicative that the SMC is inefficient. Such inefficiencies might be alleviated by increasing K or by changing the proposals. One method for estimating ESS in the context of classical SMC (Carpenter et al. 1999) can be applied here. The idea is to run PosetSMC on the same data L times for a fixed K. For scalar functions ϕ, the ratio of the variance of ϕ estimated within each run to the variance estimated across the L runs is an estimate of ESS.

PosetSMC algorithms can also benefit from improved sampling techniques that have been developed for other SMC algorithms. For example, standard techniques such as stratified sampling (Kitagawa 1996) or adaptive resampling (Moral et al. 2011) can be readily applied. Other potential improvements are reviewed in Cappé et al. (2005) and Doucet and Johansen (2009).

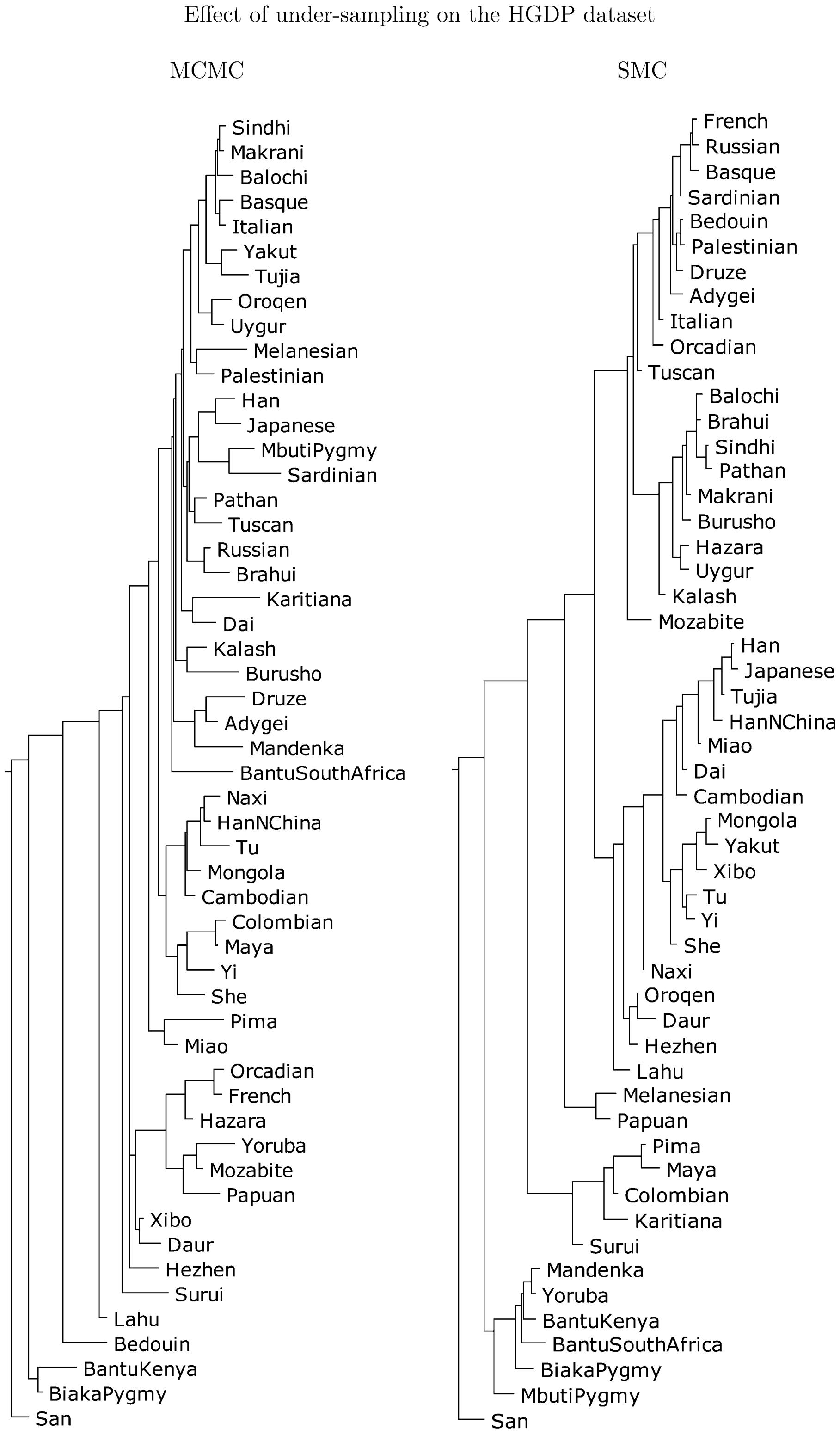

FIGURE 11. In this figure, we give a qualitative interpretation for the difference in log likelihood in Figure 10 for the consensus tree obtained from SMC with 10,000 particles and from MCMC with 10,000 iterations. Since both runs are under sampled, the higher-order groupings are incorrect in both trees, but we can see that more mid- and low-order ethnic/geographic groupings are already captured by SMC.

Note finally that the poset framework applies beyond phylogenetic tree reconstruction. Other examples of Bayesian inference problems in computational biology that may benefit from the PosetSMC framework include statistical alignment for proteins and DNA and grammatical inference for RNA.

Funding

This work was supported by the National Institutes of Health [R01 GM071749]; and the Department of Energy [SciDAC BER KP110201].

References

Altekar G., Dwarkadas S., Huelsenbeck J.P., Ronquist F. 2004. Parallel Metropolis coupled Markov chain Monte carlo for Bayesian phylogenetic inference. Bioinformatics. 20:407–415.

Andrieu C., Doucet A., Holenstein R. 2010. Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Series B. Stat. Methodol. 72:269–342.

Beaumont M.A., Zhang W., Balding D.J. 2002. Approximate Bayesian computation in population genetics. Genetics. 162:2025–2035.

Bourque M. 1978. Arbres de Steiner et réseaux dont certains sommets sont à localisation variable [PhD dissertation]. Montreal (QC): Université de Montréal.

Cannone J.J., Subramanian S., Schnare M.N., Collett J.R., D'Souza L.M., Du Y., Feng B., Lin N., Madabusi L.V., Muller K.M., Pande N., Shang Z., Yu N., Gutell R.R. 2002. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. BMC Bioinformatics. 3:2.

Cappé O., Moulines E., Rydén T. 2005. Inference in hidden Markov models. New York: Springer.

Carpenter J., Clifford P., Fearnhead P. 1999. An improved particle filter for non-linear problems. IEE Proceedings: Radar, Sonar and Navigation. Volume 146. p. 2–7.

Crisan D., Doucet A. 2002. A survey of convergence results on particle filtering for practitioners. IEEE Trans. Signal Process. 50:736–746.

Doob J.L. 1945. Markoff chains—denumerable case. Trans. Amer. Math. Soc. 58:455–473.

Douc R., Moulines E. 2008. Limit theorems for weighted samples with applications to sequential Monte Carlo methods. Ann. Stat. 36:2344–2376.

Doucet A., Freitas N.D., Gordon N., editors. 2001. Sequential Monte Carlo methods in practice. New York: Springer.

Doucet A., Johansen A.M. 2009. A tutorial on particle filtering and smoothing: fifteen years later. Cambridge (UK): Cambridge University Press.

Drummond A.J., Ho S.Y.W., Phillips M.J., Rambaut A. 2006. Relaxed phylogenetics and dating with confidence. PLoS Biol. 4:e88.

Fan Y., Wu R., Chen M.-H., Kuo L., Lewis P.O. 2011. Choosing among partition models in Bayesian phylogenetics. Mol. Biol. Evol. 28:523–532.

Felsenstein J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. Am. J. Hum. Genet. 25:471–492.

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368–376.

Felsenstein J. 2003. Inferring phylogenies. Sunderland (MA): Sinauer Associates.

Feng X., Buell D.A., Rose J.R., Waddell P.J. 2003. Parallel algorithms for Bayesian phylogenetic inference. J. Parallel. Distrib. Comput. 63:707–718.

Gelman A., Meng X.-L. 1998. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. Stat. Sci. 13:163–185.

Görür D., Teh Y.W. 2008. An efficient sequential Monte-Carlo algorithm for coalescent clustering. In: Advances in neural information processing. Red Hook (NY): Curran Associates. p. 521–528.

Griffiths R.C., Tavaré S. 1996. Monte Carlo inference methods in population genetics. Math. Comput. Model. 23:141–158.

Huelsenbeck J.P., Larget B., Swofford D. 2000. A compound Poisson process for relaxing the molecular clock. Genetics. 154: 1879–1892.

Huelsenbeck J.P., Ronquist F. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics. 17:754–755.

Huelsenbeck J.P., Ronquist F., Nielsen R., Bollback J.P. 2001. Bayesian inference of phylogeny and its impact on evolutionary biology. Science. 294:2310–2314.

Iorio M.D., Griffiths R.C. 2004. Importance sampling on coalescent histories. Adv. Appl. Probab. 36:417–433.

Keane T.M., Naughton T.J., Travers S.A.A., McInerney J.O., McCormack G.P. 2005. DPRml: distributed phylogeny reconstruction by maximum likelihood. Bioinformatics. 21:969–974.

Kimura M. 1980. A simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. 16:111–120.

Kitagawa G. 1996. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. 5:1–25.

Kong A., Liu J.S., Wong W.H. 1994. Sequential imputations and Bayesian missing data problems. J. Am. Stat. Assoc. 89:278–288.

Kuhner M.K., Felsenstein J. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. 11:459–468.

Lakner C., der Mark P.V., Huelsenbeck J.P., Larget B., Ronquist F. 2008. Efficiency of MCMC proposals in Bayesian phylogenetics. Syst. Biol. 57:86–103.

Lartillot N., Philippe H. 2006. Computing Bayes factors using thermodynamic integration. Syst. Biol. 55:195–207.

Li J.Z., Absher D.M., Tang H., Southwick A.M., Casto A.M., Ramachandran S., Cann H.M., Barsh G.S., Feldman M., Cavalli-Sforza L.L., Myers R.M. 2008. Worldwide human relationships inferred from genome-wide patterns of variation. Science. 319:1100–1104.

Liu J.S., Liang F., Wong W.H. 2000. The multiple-try method and local optimization in Metropolis sampling. J. Am. Stat. Assoc. 95:121–134.

Marjoram P., Molitor J., Plagnol V., Tavaré S. 2002. Markov chain Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. U.S.A. 100:15324–15328.

Moral P.D. 2004. Feynman-Kac formulae. New York: Springer.

Moral P.D., Doucet A., Jasra A. 2006. Sequential Monte Carlo samplers. J. R. Stat. Soc. Series B. Stat. Methodol. 68:411–436.

Moral P.D., Doucet A., Jasra A. 2012. On adaptive resampling strategies for sequential Monte Carlo methods. Bernoulli. 12:252–278.

Neal R.M. 1996. Sampling from multimodal distributions using tempered transitions. Stat. Comput. 6:353–366.

Newton M.A., Raftery A.E. 1994. Approximate Bayesian inference with the weighted likelihood bootstrap. J. R. Stat. Soc. Series B. Stat. Methodol. 56:3–48.

Paul J.S., Steinrücken M., Song Y.S. 2011. An accurate sequentially Markov conditional sampling distribution for the coalescent with recombination. Genetics. 187:1115–1128.

Penny D., McComish B.J., Charleston M.A., Hendy M.D. 2001. Mathematical elegance with biochemical realism: the covarion model of molecular evolution. J. Mol. Evol. 53:711–723.

Rannala B., Yang Z. 1996. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. J. Mol. Evol. 43:304–311.

Redelings B., Suchard M. 2005. Joint Bayesian estimation of alignment and phylogeny. Syst. Biol. 54:401–418.

Rivas E. 2005. Evolutionary models for insertions and deletions in a probabilistic modeling framework. BMC Bioinformatics. 6:63.

Robert C.P. 2001. The Bayesian choice: from decision-theoretic foundations to computational implementation. New York: Springer.

Robinson D.F., Foulds L.R. 1981. Comparison of phylogenetic trees. Math. Biosci. 33:131–147.

Saitou N., Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4:406–425.

Semple C., Steel M. 2003. Phylogenetics. Oxford: Oxford University Press.

Siepel A., Haussler D. 2004. Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. Mol. Biol. Evol. 21:468–488.

Suchard M.A., Rambaut A. 2009. Many-core algorithms for statistical phylogenetics. Bioinformatics. 25:1370–1376.

Swendsen R.H., Wang J.S. 1986. Replica Monte Carlo simulation of spin glasses. Phys. Rev. Lett. 57:2607–2609.

Tavaré S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. Lect. Math. Life Sci. 17:57–86.

Teh Y.W., Daume H. 3rd, Roy D.M. 2008. Bayesian agglomerative clustering with coalescents. Advances in neural information processing. Cambridge (MA): MIT Press. p. 1473–1480.

Thorne J.L., Kishino H., Painter I.S. 1998. Estimating the rate of evolution of the rate of molecular evolution. Mol. Biol. Evol. 15:1647–1657.

Tom J.A., Sinsheimer J.S., Suchard M.A. 2010. Reuse, recycle, reweigh: combating influenza through efficient sequential Bayesian computation for massive data. Ann. Appl. Stat. 4:1722–1748.

Xie W., Lewis P.O., Fan Y., Kuo L., Chen M.-H. 2011. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. Syst. Biol. 60:150–160.

Appendix 1

Proof of Consistency

In this section, we prove Proposition 4. The proof is similar conceptually to proofs of consistency for non-poset frameworks (Crisan and Doucet 2002; Moral 2004; Douc and Moulines 2008), with the main points of interest being the specific ways in which Assumptions 1, 2, and 3 are invoked to permit the generalization to posets. For simplicity, we present the argument for convergence in L^2 here (using $\to$ to denote convergence in L^2), assuming bounded γ_*; see Moral (2004) for many extensions, including strong laws and central limit theorems.

Proof. We start by introducing a measure $\lambda_{r,K}$ for the unnormalized distribution over the particles proposed at an SMC iteration r with K particles. From Assumption 1 and 2(b), we see that λ has the following form, for $r > 0$ and all $A \in \mathcal{F}_{\mathcal{S}}$:

$$\lambda_{r,K}(A) = \pi_{r-1,K}(\nu_{\cdot}(A)).$$

Note that in this section, we view $\pi_{r,k}$ and hence $\lambda_{r,K}$ as random variables, so we will need a notation for the limit as $K \to \infty$ of the $\lambda_{r,K}$ measures, which, as we will see shortly, has the form:

$$\lambda_r(A) = \pi_{r-1}(\nu_{\cdot}(A)),$$

where π_r is defined with the following Radon–Nikodym derivative relative to a dominating base measure μ:

$$\frac{\mathrm{d}\pi_r}{\mathrm{d}\mu}(s) = \gamma_*(s)1[\rho(s) = r].$$

The main step of the proof is to show by induction on r that:

$$\frac{1}{K}\|\pi_{r,K}\| \overset{K\to\infty}{\longrightarrow} \frac{\|\pi_r\|}{\|\pi_{r-1}\|} \text{ and} \tag{7}$$

$$\bar{\pi}_{r,K}(\phi_*) \overset{K\to\infty}{\longrightarrow} \bar{\pi}_r(\phi_*), \tag{8}$$

for any integrable $\phi_* : \mathcal{S} \to \mathbb{R}$, at a rate of C/n. This is sufficient since Equation (4) can be obtained by taking the product over Equation (7) for $r \in \{1,\ldots,R\}$, and Equation (5), by specializing Equation (8) to $r = R$ with $\phi_*(s) = \phi(s)$ for $s \in \mathcal{T}$ and $\phi_*(s) = 0$ otherwise, and by using Assumption 3(b).

The base case is a standard importance sampling argument, so we concentrate on the induction step.

Note first that by Assumptions 1 and 2(a, b), the weight $w_{r,k}$ is a deterministic function of $s_{r,k}$: Since the poset is acyclic and connected, there exists a unique $s_{r-1,k}$ covered by $s_{r,k}$, denoted by $\mathrm{pa}(s_{r,k})$ which gives all the ingredients needed to compute Equation (2). Moreover, by the Radon–Nikodym theorem and Assumptions 1, 2(a), and 3(a), this function can be written as the product of two derivatives as follows:

$$\begin{aligned} w_{r,k} = w(s) &= \gamma_*(s) \cdot \frac{1}{\gamma_*(\mathrm{pa}(s))\, q(\mathrm{pa}(s) \to s)} \\ &= \frac{\mathrm{d}\pi_r}{\mathrm{d}\mu} \cdot \frac{\mathrm{d}\mu}{\mathrm{d}\lambda_r} \\ &= \frac{\mathrm{d}\pi_r}{\mathrm{d}\lambda_r}. \end{aligned}$$

We will let the number of particles at generation r and $r' < r$ go to infinity separately, using the notation π_{r,K',K_r}.

By applying the law of large numbers, we obtain (using the fact that the weights are bounded and uncorrelated):

$$\frac{1}{K_r}\|\pi_{r,K',K_r}\| \overset{K_r\to\infty}{\longrightarrow} \bar{\lambda}_{r,K'}\left(\frac{\mathrm{d}\pi_r}{\mathrm{d}\lambda_r}\right),$$

Next, by applying the induction hypothesis to Equation (8) with $r-1$ and $\phi = \nu_{\cdot}$, we get that

$$\bar{\lambda}_{r,K'}(A) = \bar{\pi}_{r-1,K'}(\nu_{\cdot}(A)) \overset{K'\to\infty}{\longrightarrow} \bar{\pi}_{r-1}(\nu_{\cdot}(A)) = \bar{\lambda}_r(A).$$

Therefore, by setting $K = K' = K_r$, we get:

$$\begin{aligned} \lim_{K\to\infty} \frac{1}{K}\|\pi_{r,K}\| &= \bar{\lambda}_r\left(\frac{\mathrm{d}\pi_r}{\mathrm{d}\lambda_r}\right) \ (\mathrm{L}^2) \\ &= \frac{1}{\|\pi_{r-1}\|}\bar{\lambda}_r\left(\frac{\mathrm{d}\pi_r}{\mathrm{d}\bar{\lambda}_r}\right) \\ &= \frac{\|\pi_r\|}{\|\pi_{r-1}\|}, \end{aligned}$$

using $\|\lambda_r\| = \|\pi_{r-1}\|$ since the proposals $\nu_{\cdot}$ are assumed to be normalized. There is also a limit interchange argument for this step that can be formalized using the rates of convergence and the Minkowski inequality.

To prove the other induction component, Equation (8), we first use Equation (7) to rewrite the weight normalization and then proceed similarly to the above argument:

$$
\begin{aligned}
\frac{K}{K}\bar{\pi}_{r,K}(\phi_*) &= \frac{\frac{1}{K}\pi_{r,K}(\phi_*)}{\frac{1}{K}\|\pi_{r,K}\|} \overset{K\to\infty}{\longrightarrow} \frac{\|\pi_{r-1}\|}{\|\pi_r\|}\cdot\bar{\lambda}_r\left(\phi_*\cdot\frac{\mathrm{d}\pi_r}{\mathrm{d}\lambda_r}\right)\\
&= \frac{1}{\|\pi_r\|}\cdot\bar{\lambda}_r\left(\phi_*\cdot\frac{\mathrm{d}\pi_r}{\mathrm{d}\bar{\lambda}_r}\right)\\
&= \frac{1}{\|\pi_r\|}\cdot\pi_r(\phi_*)\\
&= \bar{\pi}_r(\phi_*). \qquad \square
\end{aligned}
$$

Appendix 2

Weight Updates

In this section, we simplify Equation (2) in the case of PriorPrior and PriorPost proposals on coalescent trees, showing that the updates of Teh et al. (2008) are recovered in these special cases.

In both cases, given a partial state s, a successor partial state (forest), s', is obtained from a previous partial state s by merging two trees, t_l and t_r, creating a new tree t_m (see Fig. A1). Formally, we have that $q(s \to s') > 0$ implies that there are disjoint sets $X_l = X_l(s'), X_r = X_r(s')$ such that:

$$s' = (s\backslash\{(t_l, X_l), (t_r, X_r)\}) \cup \{(t_m, X_m)\},$$

where $X_m = X_m(s') = X_l \cup X_r$, and $t_m = t_m(s')$ is an X_m-tree in which both $t_l = t_l(s')$ and $t_r = t_r(s')$ occur as subtrees. As mentioned in the main text, the new forest is also required to be strictly taller. To formalize this, consider an ordering of the speciation events in a forest s, where the heights of these events are indexed in the order they were created in the path of partial states leading to s, $0 = h_0, h_1, \ldots, h_{\rho(s)}$. Letting $\delta_i(s) = h_i - h_{i-1}$ denote the height increments, we require that for all i, $\delta_i(s) > 0$ with probability one. We denote the tree t_m by $m_s(t_l, t_r, \delta)$, where $\delta = \delta(s') = \delta_{\rho(s')}(s')$. The set of possible successor trees is denoted by $S(s)$, and the subset of successors with top increment δ is denoted by $S_\delta(s) = \{s' \in S(s) : \delta(s') = \delta\}$.

PriorPrior and PriorPost also share the same extension, γ_*, which is a product of three factors: one for the topology prior, one for the branch length prior, and one for the likelihood model $L_{\mathcal{Y}}(t)$. The first prior factor is uniform over forest topologies, and so, this first factor can be ignored since γ_* is only needed up to a normalization constant. The second prior factor is written as a distribution over height increments, $\prod_{i=1}^{\rho(s)} \Delta_i(\delta_i(s))$, where $\Delta_i(\delta)$ is an exponential density for the coalescent prior: $\Delta_i(\delta) = \mathrm{Expd}(\delta\,;\binom{|X|-i+1}{2})$. Finally, multiplying the likelihood by the prior yields:

$$\gamma_*(s) \propto \left(\prod_{i=1}^{\rho(s)} \Delta_i(\delta_i(s))\right)\left(\prod_{(t_i,X_i)\in s} L_{\mathcal{Y}(X_i)}(t_i)\right),$$

where we use the symbol $\propto$ to mean that the expression is defined up to a proportionality constant that can only depend on s via $|s|$.

We now consider the cancellations possible in Equation (2) when the PriorPrior proposal,

$$q_{\text{pr-pr}}(s \to s') \propto 1[s' \in S(s)]\Delta_{\rho(s')}(\delta(s')),$$

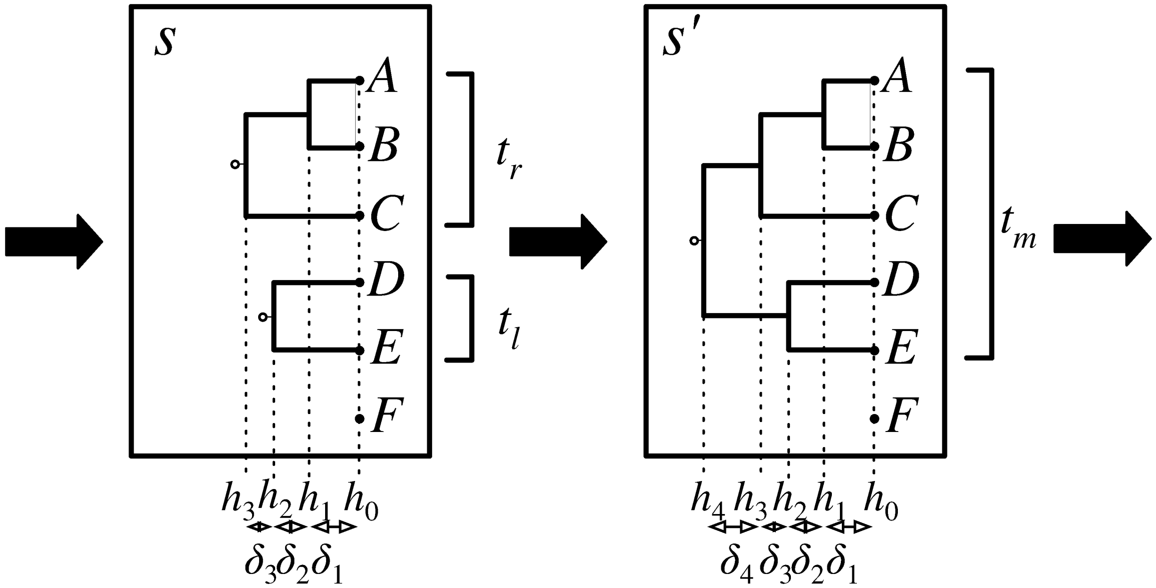

FIGURE A1. A partial state s is extended to a new partial partial state s' by merging trees t_l and t_r to form a tree t_m with height $h_4 > h_3$. In the PriorPrior proposal, t_l and t_r are chosen uniformly from the three possible pairs, whereas the height increment δ_4 is chosen from an exponential distribution. In the PriorPost proposal, δ_4 is chosen from the exponential prior and, given δ_4, the pair to merge is chosen from a multinomial with parameters proportional to the likelihood of the tree t_m.

is used:

$$w_{\text{pr–pr}}(s \to s') = \frac{\gamma_*(s')}{\gamma_*(s) q_{\text{pr–pr}}(s \to s')}$$

$$= \frac{\Delta_{\rho(s')}(\delta(s'))\, L_{\mathcal{Y}(X_m(s'))}(t_m(s'))}{L_{\mathcal{Y}(X_l(s'))}(t_l(s'))\, L_{\mathcal{Y}(X_r(s'))}(t_r(s'))\, q_{\text{pr–pr}}(s \to s')}$$

$$\propto \frac{L_{\mathcal{Y}(X_m(s'))}(t_m(s'))}{L_{\mathcal{Y}(X_l(s'))}(t_l(s'))\, L_{\mathcal{Y}(X_r(s'))}(t_r(s'))}.$$

Note that cancellations such as these are not necessary to implementing PosetSMC; indeed, they do not play a significant computational role. Our goal in presenting them was simply to make precise the connection with Teh et al. (2008).

In the PriorPost proposal, a height increment is first sampled from $\Delta_{\rho(s')}$, then, conditioning on this increment δ, the likelihood ratios

$$R_{s,\delta}(s') = 1[s' \in S_\delta(s)]\, w_{\text{pr–pr}}(s \to s')$$

are computed for all $s' \in S_\delta(s)$. These ratios are then normalized and become the parameters of a multinomial proposal over the pair of trees to coalesce:

$$q_{\text{pr–po}}(s \to s') = q_{\text{pr–pr}}(s \to s')\bar{R}_{s,\delta(s')}(s').$$

We get the following simplified form of Equation (2) for PriorPost:

$$w_{\text{pr–po}}(s \to s') = \frac{\Delta_{\rho(s')}(\delta(s'))L_{\mathcal{Y}(X_m(s'))}(t_m(s'))}{L_{\mathcal{Y}(X_l(s'))}(t_l(s'))\, L_{\mathcal{Y}(X_r(s'))}(t_r(s'))\, q_{\text{pr–po}}(s \to s')}$$

$$= \|R_{s,\delta(s')}\|.$$

Finally, we show how these weight updates can alternatively (but less transparently) be expressed as the normalization of a specific version of the peeling recursion. Let $\mathcal{Y}_j$ denote observations for site j and let $\xi_{j,t}$ denote the internal (hidden) nucleotide random variable at site j at the root of t. The recursion in question is:

$$\mathbb{P}(\xi_{j,t_m} = z|\mathcal{Y}_j(X_m))$$

$$= \frac{\mathbb{P}(\xi_{j,t_m} = z)}{C} \prod_{d \in \{l,r\}} \sum_{z'} \frac{\mathbb{P}(\xi_{j,t_d} = z' | \xi_{j,t_m} = z)\mathbb{P}(\xi_{j,t_d} = z' | \mathcal{Y}_j(X_d))}{\mathbb{P}(\xi_{j,t_d} = z')},$$

where $C = \frac{L_{\mathcal{Y}_j(X_m)}(t_m)}{L_{\mathcal{Y}_j(X_l)}(t_l)\, L_{\mathcal{Y}_j(X_r)}(t_r)}$. Therefore, the weight updates can be obtained as functions of the product of the normalizations of the top-level recursions.